Robotics – Exercise 4 – Red Team Report

Shlomi Ben-Shushan   ID: 311408264
Yiftach Neuman   ID: 208305359

Summery

In this report we will explain our implementation of two Red-Team controllers, that helped us to test, improve and evaluate our Blue-Team algorithm.

1st Red Team Controller

Algorithm Description

General Strategy: Wander the arena by driving straight until you find food or bumped into something. If you find food, continue wandering in a similar way, but if you sense base's color, drive towards it. If you bumped into something, make a hard-turn.

Sense: The forager uses the 5 not-rear RGBA cameras to sense colors and the three bumpers in its front to sense collisions.

Interpretation: By using the sensors above, the forager can understand when the friendly-nest is nearby and when it bumped into something.

Action: The followings are the forager's behavior in each state we have defined:
1. **Move:** drive straight until find food (then switch to **RTB**) or bumped into something (then make a random **turn**. Only hard-turns supported)
2. **RTB:** drive straight until sensing the nest, then drive towards it. Switch to **Move** when the food is dropped, and make a random **turn** when bumped into something
3. **Turn:** drive with angular speed only until turning timer is finished, then switch to **RTB** state if holds food, and to **Move** otherwise.

Blue vs. Red Report

Experiments: The following table describes the results of various experiments we performed in which we let the Blue-Team compete the Red-Team.

ticks	10,000 clock-ticks			100,000 clock-ticks			1,000,000 clock-ticks		
seed	Blue	Red	Winner	Blue	Red	Winner	Blue	Red	Winner
1	24	22	Blue	211	189	Blue	1019	899	Blue
2	19	22	Red	181	175	Blue	1455	1434	Blue / Tie
3	18	24	Red	199	193	Blue	1457	1350	Blue
4	18	13	Blue	182	138	Blue	835	695	Blue
5	28	22	Blue	225	188	Blue	882	833	Blue
6	24	19	Blue	189	174	Blue	1280	1135	Blue
7	27	20	Blue	181	185	Red	780	762	Blue / Tie
8	23	21	Blue	201	165	Blue	1027	839	Blue
9	17	14	Blue	179	157	Blue	1072	906	Blue
10	26	16	Blue	159	153	Blue	1067	999	Blue
11	15	18	Red	224	184	Blue	1048	997	Blue
12	22	21	Blue	204	173	Blue	937	912	Blue
13	17	11	Blue	193	129	Blue	1072	875	Blue
14	22	16	Blue	186	153	Blue	798	721	Blue
15	15	16	Red	193	171	Blue	560	528	Blue

Conclusion: The performance of the Blue-Team controller was reasonable, but not good enough under the assumption that our foragers will compete more sophisticated opponents.

2ⁿᵈ Red Team Controller

Algorithm Description

Preliminary Explanation: The first Red-Team controller let us create a good Blue-Team controller and test its performance against a close-to-naïve algorithm, but we assume that our foragers will compete against more sophisticated opponents. Thus, we decided to improve our Red-Team, in order to find a way to improve our Blue-Team.

General Strategy: Act similar to the Blue-Team algorithm (see Blue-Team Report), but without the Spiraling-Mode (because we wanted to verify that it is profitable) and without blocking the opponent's nest (because this is a feature we added later).

Sense: The forager uses the 5 not-rear RGBA cameras to sense colors and the three bumpers in its front to sense collisions (not different from the Blue-Team of previous Red-Team).

Interpretation: in this section we will describe how the forager interpret the sensing-data.

1. **Teammate Ahead:** when a front RGBA camera senses the team's color.
2. **Opponent Ahead:** when a front RGBA cameras senses the opponent team's color.
3. **Robot Ahead:** when 1 or 2 takes place (have a `true` value).
4. **Obstacle Ahead:** when frontal distance is less than 10, and there is no robot ahead.
5. **Nest Ahead:** when the front RGBA camera senses friendly-nest's color.
6. **Nest to the Right:** when the right or the right-front RGBA cameras senses friendly-nest's color, <u>and</u> the nest is not ahead.
7. **Nest to the Left:** when the left or the left-front RGBA cameras senses friendly-nest's color, <u>and</u> the nest is not ahead.
8. **Bumped into something:** when one of the front bumpers is pressed.

Action: In this section we will describe each forager state we have defined, and the behavior of the robot (the actions it will take) in each of the state.

1. **Move:** drive straight until:
    a. Found food → Switch to **RTB** state.
    b. Bumped into something → Evaluate which side is better to turn to and **hard**-turn.
    c. Teammate ahead → Evaluate which side is better to turn to and **soft**-turn.
    d. Obstacle ahead → Evaluate which side is better to turn to and **hard**-turn.
    Note that the evaluations in c. and d. are not the same.
2. **RTB:** drive straight until:
    a. Food dropped → Switch to **Move** state.
    b. Nest (or base) nearby → Drive towards it.
    c. Bumped into something → Evaluate which side is better to turn to and **hard**-turn.
    d. Obstacle ahead → Evaluate which side is better to turn to and **hard**-turn.
    Note that the evaluations in c. and d. are not the same.
3. **Soft-Turn:** drive with linear and angular speed until:
    a. Found food → Switch to **RTB** state.
    b. Turning timer-is finished → Switch to **Move**.
4. **Hard-Turn:** drive with angular speed only until:
    a. Turning-timer is finished → Switch to **RTB** if holds food, and to **Move** otherwise.

Setup() Pseudo-Code: Register for a group with `writeTeamColor()` function, define team, base, and opponent colors, w.r.t `foragingMsg.outColor` value, set initial state to "move", set other basic attributes, and start sand-timers.

Loop() Pseudo-Code: Read sensors, interpret sensing-data, and behave according to the current state, as described above.

Blue vs. Red Report

Experiments: The following table describes the results of various experiments we performed in which we let the improved Blue-Team compete the improved Red-Team.

ticks	10,000 clock-ticks			100,000 clock-ticks			1,000,000 clock-ticks		
seed	Blue	Red	Winner	Blue	Red	Winner	Blue	Red	Winner
1	14	2	Blue	72	15	Blue	715	40	Blue
2	5	0	Blue	64	0	Blue	540	33	Blue
3	11	0	Blue	97	0	Blue	676	29	Blue
4	14	0	Blue	73	0	Blue			
5	12	1	Blue	74	0	Blue			
6	14	0	Blue	82	1	Blue			
7	11	2	Blue	100	2	Blue			
8	15	1	Blue	107	0	Blue			
9	11	0	Blue	100	0	Blue	We think that's enough…		
10	8	0	Blue	62	3	Blue			
11	7	9	Red	84	11	Blue			
12	11	0	Blue	72	4	Blue			
13	8	1	Blue	60	6	Blue			
14	9	0	Blue	66	9	Blue			
15	10	0	Blue	55	0	Blue			

Conclusion: This time the results were much better, <u>although the Blue-Team faced much more sophisticated Red-Team</u>. We can see that the scores that the Blue-Team achieved are lower than in the previous experiments (which makes sense since most of the time there is one robot assigned to block the opponent's base), but the difference between the Blue-Team score and the Red-Team score is much more significant. Hence, we will use the improved Blue-Team controller in the tournament.

May the best team win!